EL978317396US

Application for United States Patent

of

Yen-Fu Chen, *et al.*

for

5   "System and Method for Automatic Information

Compatibility Detection and Pasting Intervention"

CROSS-REFERENCE TO RELATED APPLICATIONS

(CLAIMING BENEFIT UNDER 35 U.S.C. 120)

10   This application is related to U.S. Patent application number 10/455,159,

docket number AUS920030289US1, filed on June 5, 2003, by Yen-Fu Chen.

FEDERALLY SPONSORED RESEARCH

AND DEVELOPMENT STATEMENT

15   This invention was not developed in conjunction with any Federally sponsored

contract.

MICROFICHE APPENDIX

Not applicable.

INCORPORATION BY REFERENCE

20

The related patent application, U.S. Patent application number 10/455,159,

docket number AUS920030289US1, filed on June 5, 2003, by Yen-Fu Chen, is

incorporated by reference, in its entirety including figures, to the present patent

application

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] This invention relates to the arts of computer user interfaces and data

exchange between program entities such as instances of programs in a multi-tasking

5   computer system. This invention relates especially to computer methods for

transferring information from one file to another.

Background of the Invention

[0002] Modern multi-tasking computers provide a variety of user interfaces for

10   controlling multiple application programs and system functions which operate

simultaneously.   Some of the most widely used multi-tasking computer systems are

personal computers ("PC") running a multi-tasking operating system ("OS") such as

International Business Machines' ("IBM") OS/2 [TM] or AIX [TM, Microsoft

Windows [TM], and Apple Computer's MacOS [TM]. Other operating systems may

15   be used with personal computers as well as larger computers such as enterprise-class

computers, such as UNIX, Sun Microsystems' Solaris [TM], Hewlett Packard's

HP-UX [TM], and the "open sourced" LINUX.   Smaller computing platforms such

as held-held computers, personal digital assistants ("PDA"), and advanced wireless

telephones may run operating systems targeted for such hardware including Palm

20   Computing's PalmOS [TM] and Microsoft's Windows CE [TM].   Additionally, there

are many "proprietary" and less widely-used computing platforms and operating

systems which also allow users to control and run multiple programs and system functions simultaneously.

[0003] Many of these systems will use tabs, icons, windows, frames, pages and special key combinations to allow a user to switch between user interfaces ("UI") for each program and system function which is being executed, or to start or stop the execution of a program or system function. For example, in a personal computer running MS Windows [TM], the user may first start a web browser program running using several methods (e.g. double clicking an icon on the desktop, selecting the program from a Start Programs list, operating a "hot key", etc.), and then may start a document editor program using similar methods. Each program establishes a user interface such as its own "window". The user can then control a program by selecting its window using one of several available methods, such as selecting a button or icon on a command bar, activating a "task list" and selecting a program, etc. As a result, a user can start and run many programs simultaneously, periodically switching between their user interfaces to accomplish work or entertainment tasks as needed. Other computing systems provide similar basic user control capabilities, albeit with a variety of user controls to switch between programs and system functions.

[0004] Users often wish to copy or transfer information or "content" from one program or system function to another. For example, a user may be preparing an invoice for a client using a word processor program, but may also be simultaneously using a database or spreadsheet program to perform various calculations. Using "copy and paste" functions of the application programs and the operating system, the

user may select information from a source program (e.g. the spreadsheet), and "paste"

it into the destination program (e.g. the invoice being edited). Such a process is so

common place in computer users' daily lives that it is rote, albeit each user may know

several sequences of actions for several computers which he or she commonly uses

5   (e.g. one process on his home PC, another on his PDA, and another on his networked

terminal at work). These memorized methods may typically include several steps of

clicking on icons, dropping down lists, highlighting information, and using navigation

controls within program UI's.

[0005]   For example, turning to Figure 1, a "windows" style user interface is depicted

10   to illustrate a process of "copying" information from a web browser program to a

word processor file via a "clipboard" memory. In this system, each program provides

a window (2, 3, 104) which can be closed (9, 9') to end the program, maximized (8, 8')

to view the full UI for that program, or minimized (7, 7') to leave the program running

but deactivate the UI (e.g. clear the UI window from the screen). In this example,

15   these controls are located in a command bar (4, 4') along the top of the UI window,

but many other variations are known in the art.

[0006]   Each UI window also typically has navigation controls such as left panning

(15, 15'), right panning (13, 13'), and horizontal scroll (14, 14') controls, as well as up

panning (10, 10'), down panning (12, 12'), and vertical scroll (11, 11'), for viewing

20   areas of information and content not completely viewable in the UI. Information,

icons, text, graphics, etc., are shown or displayed within (16, 18) the UI window

according to the scroll and panning control settings. More recently, the term

"content" (16, 18) has been used to collectively refer to all types of information which may be displayed or presented in a user interface, including but not limited to text, graphics, still images, animated images, video, audio, and hyperlinks.

[0007]   Now suppose for the purpose of our example, the user has started a word processing program which provides a first UI window (2), and a web browser which provides a second UI window (3). Also suppose that the user is researching information on the Internet using the web browser while authoring a paper which is being edited simultaneously using the word processor.

[0008]   In this example, the user has found information (19) at a hypothetical web address (17) that he wants to "quote" in his or her paper. So, the user must first move the cursor (104) in the word processor to select an insertion point for the information, then must switch to the web browser UI, select the text (19) in the source content, operate a "copy" command in the web browser UI which copies (101) the content into a buffer (100) such as a "clipboard", switch back to the word processor UI, and operate a "paste" or "insert" command, which results in the copied content (19) being inserted into the destination document at the point of insertion (103). The user can repeat this process for many different program UI's (106).

[0009]   In some software and hardware configurations, the copy buffer may be provided within a suite of application programs which are "tightly coupled" or related. Such suites cooperate with each other in ways not possible with software programs provided by differing suppliers. In many cases, however, the operating system

provides a buffer function which is generally accessible by all programs, such as the clipboard in the MS Windows [TM] operating system.

[0010]   Also, in some situations, the original content with its original format may not be acceptable by the destination program, and as such, a specialized paste or insertion function (105) may be provided by the destination program or operating system which converts the content to a form useful by the destination program. For example, text copied from a web page may include color, size, font, style, and hyperlink reference information embedded in the base Hyper Text Markup Language ("HTML") of the source web page. However, not all word processors are able to interpret all of these special codes and identifiers, so a "paste as plain text" option may be provided by a converter or translator (105) function.

[0011]   So, to illustrate the complexity and tedious nature of such ordinary operations, we present the steps in full to accomplish this example scenario of simply transferring a block of formatted text from a web page to a word processor program, starting from a point where the user is editing the destination document in the word processor:

> (a)   navigate to the insertion point in the destination document using the word processor UI window controls (e.g. multiple clicks on scroll, panning or page up/page down keys);
>
> (b)   optionally select text or content in the destination document which is to be replaced;
>
> (c)   switch to the web browser UI window (e.g. click on an icon in a task

bar, activate a task list and pick a running web browser program, etc.);

(d)   navigate in the web browser UI window to find the text or content

desired to be transferred into the document (e.g. use panning, scrolling,

or page up/page down keys);

5   (e)   select the source content or text (e.g. click-and-drag over the content

to highlight it)

(f)   transfer the content to a copy buffer (e.g. click on "Edit" command,

select "copy" option or type Alt-E, Alt-C);

(g)   switch back to the word processor UI window (e.g. (e.g. click on a

10   icon in a task bar, activate a task list and pick a running web browser

program, etc.); and

(h)   operate a "paste" command in the word processor UI window (e.g.

click on "Edit" command, select "paste" option or type Alt-E, Alt-P).

15   **[0012]**   Each of these operations may actually require several steps (clicking,

scrolling, selecting, typing, etc.), so this minimal process may represent 7 to 25 actual

user actions. This process must be repeated for each block of text or content to be

transferred from multiple program UI windows (106), and additional steps may be

necessary to achieve a "special paste", as described above. Also, if the same text or

20   content is to be inserted into the destination document for file at multiple locations, the

last few operations of this process (h) in addition to some navigation actions must by

performed by the user.

[0013]   As a result, consolidating information from multiple sources of information may be extremely tedious, frustrating, and tiresome using the currently available methods and apparatuses provided in such computing systems.   Some systems may provide notably more "user friendly" or intuitive methods, while other systems are much more difficult and "clunky" to use.

[0014]   Turning to Figure 2, this process is generalized.   Starting at a point or time (21) when the user is actively working with the destination program UI, the user must navigate (22) within the present document, file, or other computer resource to a point where the content insertion is to be made, including selecting any content which is to be replaced.   Then, the user must switch (23) to the UI of the first source of information, navigate (24) to the first source content to be transferred, select that content, and operate (25) a copy or cut control in the first source UI.

[0015]   Next, the user must switch (26) back to the destination UI, and operate (27) an insert or paste command in that UI. If (28) the user wants to insert or paste that content into multiple destinations, the user must navigate (29) to each destination and operate (27) the paste or insert command in the destination program UI, until all insertions have been made for that source information.

[0016]   If (200) the user desires to transfer information from other points in the same source, or from other sources, the user must repeatedly switch (201) to a source UI, navigate to a source content point, select source information, operate (25) a copy or cut operation, switch (26) back to the destination UI, and paste or insert (27) the content, until all information has been transferred.

[0017]   Implied, but not shown in detail, in this generalization of the process can be multiple user actions for each general step.   Optionally, options such as conversion of the content may be necessary, which requires additional user actions (e.g. the "paste as plain text" example).

5   [0018]   So, it is not inconceivable that in the course of authoring a paper using a word processor and information from several sources, the user may have to perform hundreds of tedious actions, commands, selections, navigation operations, etc.

[0019]   In this paradigm, certain conventions have evolved into place which only moderately simplify or reduce the burden of such operations.   For example,

10   performing a "cut" operation usually deletes the selected source content from the source file, and places a copy of it into the transfer buffer, sometimes overwriting the current contents of the transfer buffer.   A "copy" operation typically leaves the selected information unchanged in the source and only places a copy of the information in the transfer buffer.   Additionally, in the destination UI, a "paste" or

15   "insert" command may copy the contents of transfer buffer to a selected point in the destination document or file, leaving a copy in the transfer buffer for additional pastes or insertions.

[0020]   In some programs, a "paste special", "import from clipboard", or similar command may be available with several conversion options to perform a minimal

20   conversion process on each transfer.   However, even though the user may be performing the same "paste special" command over and over, the typical UI does not

memorize or "learn" this process, so the user is forced to respond to a number of redundant options and dialogs on each paste operation.

[0021]   The same user interface conventions are followed by many computer systems not only for content or information within a computer resource such as text and

5   graphics within a file, but also for resources (e.g. files, shortcuts, icons, mappings, etc.) within a computing environment (e.g. file system, directories, folders, etc.).   For example, when working with a MS Windows [TM] operating system and running the Windows Explorer program, a user may select a file, directory or folder to move, execute an "Edit - Cut" command sequence, navigate to another directory or drive,

10   and execute an "Edit - Paste" command to move the selected resource to the new destination.   Similarly, by selecting the source resource, executing a copy command, and then executing a paste command to one or more destinations, the original resource is not changed but copies of it are deposited at the destination points.   Further, by selecting and copying a source resource, then selecting a destination resource,

15   replacement of the destination resource may be accomplished.

[0022]   The invention described in the related patent application entitled "System and Method for Content and Information Transfer Between Program Entities", filed by Yen-Fu Chen, *et al.*, and incorporated herein, addressed this problem to a large degree.   The related invention enables users to preselect or pre-designate one or more

20   destination content area(s) or points, and then to only perform the highlighting action in the source files or documents to produce copy/paste results without any additional mouse clicks or touching the keyboard.

[0023]   According to one aspect of the related invention, the user initially designates one or more insertion points or replacement areas for receiving transferred content in a destination UI, then switches to a source UI, highlights content to be transferred, and the system performs the copy or paste operation without any additional user action,

5   including any conversion as may be necessary.

[0024]   Then, the user may simply navigate to another point in the same source, or switch to another source UI, select content, and it will be "pasted" automatically into the destination without the user having to switch back to the destination UI or perform any additional paste or insert command actions.

10   [0025]   However, as source information may or may not be completely compatible with destination environment, the user, even when using the related invention, may be required to make certain tedious and inconvenient operations to perform the information transfer usefully.

[0026]   For example, consider a situation where the source information is a

15   graphically rich section of a web page, including text and color photographs. Further assume for the purposes of this example that the color photographs are stored in the source document as Joint Photographic Experts Group ("JPEG") format data objects. If the user attempts to transfer this selected information (e.g. text + photograph) to a destination which does not support JPEG photographs, such as a text-only editor, a

20   problem may arise that cannot be completely handled by the related invention -- e.g. what to do with the photograph. In some other cases, destination editor may be able

to handle other format images, such as Graphic Interface File ("GIF") images, but not

JPEG images.

[0027]   The user may, after realizing this and with sufficient technical expertise, find

a way to export the image and save it to a separate file, use another tool to convert the

5   JPEG image to a GIF image, and then use the related invention to transfer the GIF

image to the destination document. This process, however, requires the user to have

the expertise necessary to make such an export and conversion, as well as to have the

extra tool to perform the conversion. Further, if this is a task to be repeated often,

such as cutting and pasting a considerable amount of information in this manner, it

10   becomes tedious, tiresome, and error-prone.

[0028]   Therefore, there is a need in the art for a system and method to provide a

configurable automatic source-to-destination compatibility detection and conversion

system and method for transferring information from one or more source environments

to a destination environment.

15

SUMMARY OF THE INVENTION

[0029]   The existing approaches for transferring information and content between

multi-tasked programs in a computer system can certainly create confusion, are

tedious, time consuming, and frustrating. The present invention enables users to

5   preselect or pre-designate one or more destination content area(s) or points, and then

to only perform the highlighting action in the source files or documents to produce

copy/paste results without any additional mouse clicks or touching the keyboard.

[0030]   Additionally, the present invention automatically intercepts the transfer of the

information to the destination, detects information compatibility during information

10   transfer (e.g. during a "pasting from buffer" operation"), determining if the information

is compatible with the destination document, environment, or data construct. If the

information is not compatible, the transfer is suspended, and the user is provided an

opportunity to define a new rule, process, or macro regarding how to handle the

incompatible information.   Options include invoking a converter or translator, if one is

15   available on the present system. Another option is to export the incompatible

information to a separate data construct such as a separate file, and to provide a note

or annotation in the destination file as to the removal of the information and its final

location.

BRIEF DESCRIPTION OF THE DRAWINGS

[0031] The following detailed description when taken in conjunction with the figures presented herein provide a complete disclosure of the invention.

[0032] Figure 1 illustrates the typical user interfaces and system actions of "cutting and pasting" information from one application program to another.

[0033] Figure 2 shows the typical logical process in a general manner for transferring content from one application program or system function to another.

[0034] Figure 3 depicts a generalized computing platform architecture, such as a personal computer, server computer, personal digital assistant, web-enabled wireless telephone, or other processor-based device.

[0035] Figure 4 shows a generalized organization of software and firmware associated with the generalized architecture of Figure 1.

[0036] Figure 5 illustrates the logical method of the present invention.

[0037] Figure 6 depicts details of the logical process of handling incompatible elements of information.

[0038] Figure 7 shows an example of isolation of an incompatible element, in this case a graphic element.

DESCRIPTION OF THE INVENTION

[0039] The present invention is preferably realized as an enhancement of the related invention, but may be alternately realized in some embodiments as a stand alone tool or as an improvement to other products and application program. As such, the

5   present invention will be described in detail relative to the details of the related invention. It is within the skill of those in the art to adapt the present invention to other embodiments, when provided with the description contained herein.

[0040] According to one aspect of the invention, the user initially designates one or more insertion points or replacement areas for receiving transferred content in a

10   destination UI, then switches to a source UI, highlights content to be transferred, and the system performs the copy or paste operation without any additional user action, including any conversion as may be necessary.

[0041] Then, the user may simply navigate to another point in the same source, or switch to another source UI, select content, and it will be "pasted" automatically into

15   the destination without the user having to switch back to the destination UI or perform any additional paste or insert command actions.

[0042] The information contained in the transfer buffer may be concatenated as source selection operations are made instead of replacing the contents of the buffer, such that a user may "collect up" information from multiple source points before a

20   single insertion or paste is made

[0043] Content and information conversion rules may be established, either by the programs themselves, by the user, or by both, which will control the automatic

conversion of the information from one format to another prior to inserting it into the

destination file or document. Using these rules, the invention automatically detects

when part or all of content being pasted is incompatible with the destination document,

file or data construct. This invokes a user interface which allows the user to establish

5   a new rule for conversion or handling of the incompatible content, if no handling rule

exists. For future translations of similar information, the invention can then handle the

conversion or handling of the information automatically according to the user's

established rules.

10   <u>Suitable Computing Platforms</u>

[0044] The present invention is preferably realized as computer-executable code

such as compiled software, scripts, or portable programs, in conjunction with existing

application programs, operating systems, and computer hardware. For purposes of

completeness, we first present a generalized view in Figures 3 and 4 of typical

15   computer hardware and software which is suitable for realization of our invention.

Many variations of these computer platform details may be made without departing

from the scope of the invention, as will be readily recognized by those skilled in the

art.

[0045] Common computing platforms such as personal computers, web servers, and

20   web browsers, as well as proprietary computing platforms, may be used in realization

of the present invention. These common computing platforms can include, but are not

limited to, personal computers as well as portable computing platforms, such as

personal digital assistants ("PDA"), web-enabled wireless telephones, and other types of personal information management ("PIM") devices.

[0046] Therefore, it is useful to review a generalized architecture of a computing platform which may span the range of implementation, from a high-end web or

5   enterprise server platform, to a personal computer, to a portable PDA or web-enabled wireless phone.

[0047] Turning to Figure 3, a generalized architecture is presented including a central processing unit (31) ("CPU"), which is typically comprised of a microprocessor (32) associated with random access memory ("RAM") (34) and read-only memory

10   ("ROM") (35). Often, the CPU (31) is also provided with cache memory (33) and programmable FlashROM (36). The interface (37) between the microprocessor (32) and the various types of CPU memory is often referred to as a "local bus", but also may be a more generic or industry standard bus.

[0048] Many computing platforms are also provided with one or more storage

15   drives (39), such as a hard-disk drives ("HDD"), floppy disk drives, compact disc drives (CD, CD-R, CD-RW, DVD, DVD-R, etc.), and proprietary disk and tape drives (e.g., Iomega Zip [TM] and Jaz [TM], Addonics SuperDisk [TM], etc.). Additionally, some storage drives may be accessible over a computer network.

[0049] Many computing platforms are provided with one or more communication

20   interfaces (310), according to the function intended of the computing platform. For example, a personal computer is often provided with a high speed serial port (RS-232, RS-422, etc.), an enhanced parallel port ("EPP"), and one or more universal serial bus

("USB") ports. The computing platform may also be provided with a local area network ("LAN") interface, such as an Ethernet card, and other high-speed interfaces such as the High Performance Serial Bus IEEE-1394.

[0050]   Computing platforms such as wireless telephones and wireless networked

5   PDA's may also be provided with a radio frequency ("RF") interface with antenna, as well. In some cases, the computing platform may be provided with an infrared data arrangement (IrDA) interface, too.

[0051]   Computing platforms are often equipped with one or more internal expansion slots (311), such as Industry Standard Architecture ("ISA"), Enhanced Industry

10   Standard Architecture ("EISA"), Peripheral Component Interconnect ("PCI"), or proprietary interface slots for the addition of other hardware, such as sound cards, memory boards, and graphics accelerators.

[0052]   Additionally, many units, such as laptop computers and PDA's, are provided with one or more external expansion slots (312) allowing the user the ability to easily

15   install and remove hardware expansion devices, such as PCMCIA cards, SmartMedia cards, and various proprietary modules such as removable hard drives, CD drives, and floppy drives.

[0053]   Often, the storage drives (39), communication interfaces (310), internal expansion slots (311) and external expansion slots (312) are interconnected with the

20   CPU (31) via a standard or industry open bus architecture (38), such as ISA, EISA, or PCI. In many cases, the bus (38) may be of a proprietary design.

[0054]   A computing platform is usually provided with one or more user input

devices, such as a keyboard or a keypad (316), and mouse or pointer device (317),

and/or a touch-screen display (318). In the case of a personal computer, a full size

keyboard is often provided along with a mouse or pointer device, such as a track ball

5   or TrackPoint [TM]. In the case of a web-enabled wireless telephone, a simple keypad

may be provided with one or more function-specific keys. In the case of a PDA, a

touch-screen (318) is usually provided, often with handwriting recognition capabilities.

[0055]   Additionally, a microphone (319), such as the microphone of a web-enabled

wireless telephone or the microphone of a personal computer, is supplied with the

10   computing platform. This microphone may be used for simply reporting audio and

voice signals, and it may also be used for entering user choices, such as voice

navigation of web sites or auto-dialing telephone numbers, using voice recognition

capabilities.

[0056]   Many computing platforms are also equipped with a camera device (3100),

15   such as a still digital camera or full motion video digital camera.

[0057]   One or more user output devices, such as a display (313), are also provided

with most computing platforms. The display (313) may take many forms, including a

Cathode Ray Tube ("CRT"), a Thin Flat Transistor ("TFT") array, or a simple set of

light emitting diodes ("LED") or liquid crystal display ("LCD") indicators.

20   [0058]   One or more speakers (314) and/or annunciators (315) are often associated

with computing platforms, too. The speakers (314) may be used to reproduce audio

and music, such as the speaker of a wireless telephone or the speakers of a personal

computer. Annunciators (315) may take the form of simple beep emitters or buzzers, commonly found on certain devices such as PDAs and PIMs.

[0059]   These user input and output devices may be directly interconnected (38', 38") to the CPU (31) via a proprietary bus structure and/or interfaces, or they may be interconnected through one or more industry open buses such as ISA, EISA, PCI, etc.

[0060]   The computing platform is also provided with one or more software and firmware (3101) programs to implement the desired functionality of the computing platforms.

[0061]   Turning to now Figure 4, more detail is given of a generalized organization of software and firmware (3101) on this range of computing platforms. One or more operating system ("OS") native application programs (43) may be provided on the computing platform, such as word processors, spreadsheets, contact management utilities, address book, calendar, email client, presentation, financial and bookkeeping programs.

[0062]   Additionally, one or more "portable" or device-independent programs (44) may be provided, which must be interpreted by an OS-native platform-specific interpreter (45), such as Java [TM] scripts and programs.

[0063]   Often, computing platforms are also provided with a form of web browser or microbrowser (46), which may also include one or more extensions to the browser such as browser plug-ins (47).

[0064]   The computing device is often provided with an operating system (40), such as Microsoft Windows [TM], UNIX, IBM OS/2 [TM], LINUX, MAC OS [TM] or

other platform specific operating systems.   Smaller devices such as PDA's and

wireless telephones may be equipped with other forms of operating systems such as

real-time operating systems ("RTOS") or Palm Computing's PalmOS [TM].

[0065]   A set of basic input and output functions ("BIOS") and hardware device

5   drivers (41) are often provided to allow the operating system (40) and programs to

interface to and control the specific hardware functions provided with the computing

platform.

[0066]   Additionally, one or more embedded firmware programs (42) are commonly

provided with many computing platforms, which are executed by onboard or

10   "embedded" microprocessors as part of the peripheral device, such as a micro

controller or a hard drive, a communication processor, network interface card, or

sound or graphics card.

[0067]   As such, Figures 3 and 4 describe in a general sense the various hardware

components, software and firmware programs of a wide variety of computing

15   platforms, including but not limited to personal computers, PDAs, PIMs, web-enabled

telephones, and other appliances such as WebTV [TM] units.

Logical Processes of the Invention

[0068]   We now turn our attention to disclosure of the present invention relative to

20   the processes and methods preferably implemented as software and firmware on such a

computing platform. It will be readily recognized by those skilled in the art that the

following methods and processes may be alternatively realized as hardware functions, in part or in whole, without departing from the spirit and scope of the invention.

[0069]   In general, the present invention provides the user the ability to follow this process:

5   1. Declare destination content area for pasting or insertion in a destination UI;

           2. Enable highlighting or selecting content to automatic copy selected information and store it in memory (e.g. buffer/clipboard) for future usage, as well as to automatically insert or paste the selected information into the pre-designated destination area;

10   3. Switch to source UI's, and simply highlight or select the desired content portion(s) which triggers dynamic insertion/concatenation into the declared content field.

           4. Receive an automatic notification that a portion of the highlighted source information is not compatible with the destination; and

15   5. Define new rule or process for handling the incompatible content such as conversion, isolation and annotation, such that future similar situations are handled automatically according to the user's preferences.

20 [0070]   Using the present invention, advantages over presently available methods and user interfaces are:

           1. User-friendly: the invention utilizes maneuvering techniques with which

users are already familiar, such as double mouse clicks, dragging to

highlight, etc.;

2. Convenient: the invention provides a user a new way of selecting

destination for paste operations, and offers a quicker way of copy/paste

by eliminating keyboard strokes, mouse clicks, navigation and

toggling between different program windows and UI's.

3. Intuitive: Users can learn or "discover" that in this new mode, content

highlighting triggers copy/paste effortlessly, immediately seeing the

results of the highlighting action, and proceeding to experiment with

various ways of applying and exploring this invention.

4. Time-Saving: Reduces time to compile information from multiple sources

by making copy-paste user interface methods much more efficient.

5. Increases productivity: By reducing fatigue and tediousness in application

program user interfaces, users are allowed to produce more accurate

results with less effort in less time.

[0071]   Turning to Figure 5, a first logical process of the invention is shown at a high

level. This method may be implemented as compiled code, scripts, or interpretable

code (e.g. Java, applets, servlets, etc.) in full or part within application programs,

operating systems, server suites, utility programs, or proprietary code or hardware.

[0072]   While (21) creating or editing a computer resource such as a computer file or

document, when a user wishes to insert or copy content from a source destination to

the resource being edited, the user may execute the method (50) shown.   The source

or sources may include points and areas within UI's of other programs and system

functions, as well as within the destination resource which is being created or edited.

[0073]   Initially, the user may navigate (51) to a first insertion point and optionally

5   highlight an insertion area to designate a first "paste" destination while in the current

(destination) UI.   If (52) more than one insertion points or areas are desired, the user

may navigate to and select (53) additional destinations while still in the destination UI.

Insertion points can be marked in a variety of ways, including placement of a graphical

icon such as a caret (" ^ ") at the insertion point(s) or other suitable icon.

10   Replacement destination areas can be marked in a variety of ways, including

highlighting the selected text, graphics, information, or content in a special pattern,

color, or both, or by making special changes to the selected information itself (e.g.

changing the text to a different color, underlining the text, etc.).   Preferably, a user

interface or dialog box is also provided to allow the user to manage destination

15   insertion points and replacement areas, such as to create additional points and areas,

and to remove points and areas.

[0074]   When all destination points and areas have been designated, the user enables

(54) the automatic content transfer function of the invention through an appropriate

user control such as right-clicking with the mouse to bring up a dialog box, selecting

20   an option from a drop-down list, or operating a special icon, key or key sequence.

The user may be allowed to select advanced options, such as rules for concatenation

or content conversion, or whether cutting (e.g. source deletion) is performed as well as copying on each subsequent selection operation.

[0075]   Next, the user switches (55) to the first UI from which source information is to be transferred, navigates to the first source point, and highlights or selects (56) source content.

[0076]   Compatibility handling is performed (501), described in more detail in the following paragraphs, and the compatible information is then automatically copied into the transfer buffer, and subsequently pasted (58) into the designated destination point(s) and area(s), without need to switch back to the destination UI.

[0077]   If (57) more source content is to be transferred, the user may continue to navigate to additional points and to select (56) additional content for automatic pasting into the destination resource from within the current UI, or the user may switch (500, 501) to another UI and proceed with selecting (56) for automatic pasting (58) into the destination resource.

[0078]   If the concatenation option is selected, upon selection of a content block or area, it is appended to the existing contents of the transfer buffer, and the user is given the option to perform the automatic pasting into the destination point(s) and area(s), without need to switch back to the destination UI.

[0079]   If the source cutting operation is selected, upon selection of a content block or area, it is deleted from the source resource (assuming the user has the rights to modify the resource), placed in the transfer buffer, and pasted, concatenated or converted as determined by the other options.

[0080]   When (57) all source information has been copied, the information transfer

operation is complete (59), and the user returns to the initial (e.g. destination) UI to

continue editing or creating (21') the computer resource.

[0081]   As an example and comparison now between the user control method

typically employed and the user control method provided by the present invention, we

present the following case. On a MS Windows]TM] operating system, a user who

has several windows opened intends to compose an email using a web based mail

program. Further, the user plans to generate the content of this email from the other

opened windows by using the copy/paste functions, specifically from Microsoft Word

[TM] word processor and from Lotus Notes [TM] application programs. Suppose

that the email message is to be created from a paragraph in MS Word and another

from Lotus Notes respectively. So, traditionally what the user would be required to

do in order to accomplish this task in the shortest time frame is the following:

1.   open the email composer window;

2.   switch to the window for MS Word using Alt-Tab;

3.   highlight the desired text in the MS Word document;

4.   hit Ctrl-C to copy the selected text into the clipboard;

5.   hit Alt-Tab until toggled to the email composer window;

6.   click on the text area or point where the copied paragraph is to be

placed;

7.   hit Ctrl-V or select Edit-Paste to insert the text copied from MS Word;

8.   switch to the Lotus Notes window using Alt-Tab;

9.   highlight or select the desired text in Notes;

10.   hit Ctrl-C to copy the text to the clipboard;

11.   hit Alt-Tab to switch back to the email composer window;

12.   place the cursor on the end of the first inserted paragraph;

5   13.   hit Ctrl-V or select Edit-Paste with the mouse or keyboard to append

text from the Notes window to the initially inserted content; and

14.   switch back to the email composer window using Alt-Tab.

[0082]   Using the invention, the method is shortened significantly to achieve identical

10   result without ever having to touch the keyboard:

1.   open the email composer window, place cursor at desired insertion

point;

2.   enable auto-pasting and designate an insertion point or replacement

area (e.g. by right clicking in the email text area or by double clicking

15   on the left mouse button while in the email message text area),

selecting an auto-paste option (assume no concatenation, no

conversion and cutting options are selected)

3.   switch to the MS Word window using Alt-Tab;

4.   highlight the first block of desired text causing the selected

20   text to be automatically inserted in the email composer

window;

5.   switch to the Lotus Notes window using Alt-Tab;

6.   highlight the second block desired text causing the selected

text to be automatically inserted in the email composer

window following the first block; and

7.   switch back to the email composer window using Alt-Tab.

5

[0083]   There are alternative implementations available to declare destination context

area. For example, consider the situation where the email message is created from

many windows or many sections of a few windows, the number of toggles between

program windows and highlighting/Ctrl-C/Ctrl-V operations could cause intense

10   frustration and stress on the user.   In systems running operating system which are not

completely stable, such window or frame switching may also increase the possibility of

an OS lock up or crash.

[0084]   Similar to the email example, this invention can be used by helpdesk agents

filling out problem descriptions from known issues or solutions, users completing a

15   form based on known information (creating resume, registration, etc.), and even

programmer and software testers attempting to write new code or test cases by using a

template or reusing available code.

[0085]   There are plentiful situations where an efficient way of copying content and

inserting it into one or more desired locations can significantly improve productivity,

20   save time, reduce overall operational cost, and radically increase user satisfaction.

[0086]   In an alternate embodiment of the present invention, each selection of source

content may be followed by addition of a specified delimiter between the next selection

of source content. For example, in the email composition scenario previously

discussed, a delimiter such as a space, carriage return, HTML horizontal line, or text

note may be automatically added between the MS Word paragraph and the Lotus

Notes paragraph. In a first advanced implementation of this option, a descriptive

5   annotation may be appended before or after the content, such as:

                        <copied from MS Word: c:\my_files\document.doc>

     and:

10   <copied from Lotus Notes: e:\network_backups\advice.lwp>

[0087] These notations can include web site addresses, as well, and optionally time

and date information:

15   <copied from www.help.ibm.com\laptops\upgrades on 3-15-2003>

[0088] Alternate embodiments of this notation include, but are not limited to,

annotations made in eXtensible Markup Language ("XML"), and possibly including a

time stamp such as a local clock and calendar value, or a Coordinated Universal Time

20   ("UTC") value.

[0089] According to another alternative aspect of the present invention, the

destination areas and points may be saved in association with a computer resource

(e.g. a specific file), or as a named template. These insertion points and areas can then be recalled so that new information from multiple sources can be quickly and routinely compiled. This function is particularly useful for creating news letters, status reports, and the like.

5   [0090]   As will be recognized by those skilled in the art, the use of text as the content copied from a source to a destination in the foregoing examples is not meant to imply that the present invention is limited to such content types. In fact, it may be used to transfer content of any format which is compatible with the transfer buffer implementation. For example, video clips, graphics, hyperlinks, audio clips, and the

10   like may be handled by the invention provided that the transfer buffer is implemented in a manner compatible with storing such information.

[0091]   Another desirable but not critical aspect of the invention is to provide an "undo" function, with or without an action log. By keeping track of the source to destination transfers, and especially the source deletions when "cutting" is selected, the

15   user may reverse any operation previously made. The tracking record of actions taken can preferably be produced in a human readable form, which is useful in version tracking, revision control, and similar activities.

Compatibility Handling

20   [0092]   Turning to Figure 6, details of compatibility handling (501) according to the present invention are shown. A set of compatibility rules, if already defined, are consulted initially to determine (60) if all the elements of the selected source content

are compatible with the destination.   For example, Table 1 shows a basic set of

compatibility rules which can be implemented in a text file for a particular hypothetical

destination document editor called "JotPro". For this example, also assume that

JotPro can receive plain text, richly formatted text (e.g. text with font, size, color and

5   other attributes), Microsoft Windows [TM] text, ANSI text, ASCII text, as well as

Tagged Interchange Format ("TIF") and Bit Map ("BMP") images, but not JPEG, GIF

or Windows Metafile ("WMF") images.   Also, our hypothetical word processor has

its own native text format, JotText, and native image format JotPix. Further assume

that JotPro documents can include hyperlinks.

10

- -

Table 1. Example "JotPro" Compatibility Rules

- -

```
<compatibility_rules>

15              <destination> Jotpro </destination>

                <destination_versions> all </destination_versions>

                <text_compatibility>

                    plain; ascii; ansi; MSWin; RTF; JotText

                </text_compatibility>

20              <graphic_compatibility>

                    tif; tiff; bmp; JotPix;

                </graphic_compatibility>
```

</compatibility_rules>

- -

[0093]   Table 1 is expressed in a markup-style language, such as eXtensible Markup

Language ("XML").   Other formats and language are available to implement such as

set of compatibility rules, including but not limited to plain text, macro languages,

tables, databases, etc.

[0094]   So, each element (e.g. paragraph or sentence of text, graphic image, video

clip, audio clip, animation, etc.) is checked (60) the list of compatibility rules, and if it

is not found to be compatible, it is marked for conversion.   If (62) all elements are

compatible, then transfer of the unmodified source content (all elements) is allowed

(63).

[0095]   For each incompatible element, a set of conversion rules (65) is consulted

(64) to determine how to handle the element.   Table 2 shows an example set of

conversion rules expressed in a markup-style language.

- -

Table 2. Example "JotPro" Conversion Rules

- -

5   `<conversion_rules>`

            `<destination> Jotpro </destination>`

            `<destination_versions> all </destination_versions>`

            `<rule>`

                `<element_source>"Lotus WordPro" </element_source>`

10   `<element_type> text </element_type>`

                `<action> invoke WP2JP_text.exe </action>`

            `</rule>`

            `<rule>`

15   `<element_source>"Lotus Presentation" </element_source>`

                `<element_type> vector graphics </element_type>`

                `<action> invoke WP2JP_graphics.exe </action>`

            `</rule>`

20   `<rule>`

                `<element_source> all </element_source>`

                `<element_type> JPEG; JPG </element_type>`

<action> invoke JPG2JP_graphics.exe </action>

</rule>

</conversion_rules>

5   -

[0096]   In this example set of rules, text which is being pasted from the Lotus

WordPro [TM] word processor's native format into JotPro will be first imported or

converted using a program specified as "WP2JP_text.exe".   Likewise, vector graphics

10   being pasted from Lotus Presentation will be converted using a program specified as

WP2JP_graphics.exe".   Additionally, any graphics elements of the JPEG type (e.g.

JPG or JPEG) will be converted using a program specified as "JPG2JP_graphics.exe".

It is important to note that the example program names given here are hypothetical as

well, and any other suitable macro, program, applet, servlet, or other suitable process

15   name or handle can be used in their place.

[0097]   If (66) no applicable rule for an incompatible element is found in the

conversion rules (65), then a user interface is preferably invoked to allow the user to

define a new rule or action for handling the type of incompatible element.

[0098]   When conversion rules for all incompatible elements have been found or

20   defined, the various actions or conversion operations are performed (68), such as

importing, converting, or isolating an element.   Importing and converting actions are

performed when a translator or "import filter" is available which is capable of

converting the incompatible element to a compatible element, such as converting rich

text to proprietary text, or converting a graphics element from JPEG to GIF, etc.

[0099]   Isolation is a special type of handling which allows incompatible elements to

be "separated" from the source content and stored in a separate data construct, such as

5   a file on the system hard drive, or a file in system memory.   For example, Figure 7

shows an isolation process wherein some example text "Qwerty utrew zxf" has been

selected along with a graphic image (shown in the solid box), such as a GIF image.

However, the compatibility handler has been configured to isolate this particular type

of graphic image, so it is directed towards a file called "image-01.gif" stored as an

10   isolated element (69), and only the compatible text "Qwerty utrew zxf" is transferred

to the destination (103). Optionally and preferably, an annotation indicating the

isolation and location of the incompatible element, as shown.   Preferably, if the

destination file can include hyperlinks, the annotation is a hyperlink to the actual

isolated element, thereby allowing the user to easily and quickly find the isolated

15   element.

Rule Management User Interface

[0100]   As previously disclosed, when an element is encountered in the source

content for which there is no compatibility rule or no conversion rule, a user interface

20   is preferably invoked which allows a user to:

        (a)   define a new type of element for a given destination with which the

              destination is compatible;

(b)　delete a type of element for which a given destination is incompatible

(this allows a user to selectively block certain element types from

transfer, and optionally invoke a conversion rule on that element);

(c)　define a new conversion rule for a given element type and destination

5　　　　　　　including simply blocking the transfer of the element, converting

the element and transferring it, isolating the element, isolating the

element and transferring an annotation, and isolating the element and

transferring a hyperlinked annotation.

(d)　modifying or deleting an existing conversion rule.

10

[0101]　Well-known user interface elements, such as drop-down boxes, modifiable

lists, browsing windows (for finding converters and executable translators), radio

buttons, etc., can be used to realize such a user interface which creates, edits and

manages the rules, such as those shown in Tables 1 and 2.　The compatibility rules

15　and the conversion rules may be stored separately, or together, according to the

particulars of the alternative embodiment of the invention.

Conclusion

[0102]　As such, the present invention may be realized in a variety of forms,

20　programming languages, methodologies, and operating systems on a variety of

computing platforms without departure from the spirit and scope of the present

invention.　A number of example embodiment details have been disclosed as well as

-37-

optional aspects of the present invention in order to illustrate the invention, but which do not define the scope of the invention. Therefore, the scope of the present invention should be determined by the following claims.